TRITON/BAAN IV

BAAN Translation Manual

Installation Guide U7062A US

BAAN

Document information

Document

Document code   : U7062A US
Document group   : Development Documents
Document title   : BAAN Translation Manual
Edition   : A
Month modified   : October 1997

Check the BaanDoc database to see if this document is still a valid edition.

Table of contents

About this document

This manual explains how to translate BAAN software and help in BAAN Tools. It describes the entire translation procedure from preparation and planning through testing.

Chapter 1 addresses the translation project.

Chapter 2 addresses translation project preparation in terms of human resources and knowledge build-up.

Chapter 3 addresses the translation environment and the way it is prepared for the translation project. Most of these steps involve BAAN application administrator activities, rather than specific translator activities.

Chapter 4 addresses the copying of the software prior to translation.

Chapter 5 addresses the translation of the software and the help texts inside the application

Chapter 6 addresses the translation of the software and the help texts in an export file.

Chapter 7 addresses the verification of the software and the help texts.

Chapter 8 addresses the procedure to make the translations available to the end user.

Chapter 9 addresses the validation and run-time testing of software and help.

Chapter 10 addresses the way to handle updates, by using a Translation Memory.

Various appendices deal with Unix and vi, checklists, and other convenient issues.

For technical matters such as software installation, customization, and releasing translated software packages, refer to the BAAN Tools User Manual and the BAAN Tools Installation Manual.

1. Introduction

1.1 Objectives

Depending on the software to be translated and the (commercial) goals, the translation project may have different objectives:

- For a demo version, the objective is to translate only the software components visible to the end user, and the most important modules.

- For a full translation, the objective is to translate entire packages/modules of one version, either manually or by using translation tools.

- An update translation involves the translation of the net changes from one version to the next (TRITON 3.0 to BAAN IV) or from one release to the next (BAAN IV b to BAAN IV c).

1.2 Approach

The standard approach to translate a complete software package consists of two consecutive projects: one for the software and one for the on-line help texts. This two-phase approach is especially advised when a large software product is to be translated for the first time. If you first translate the software, that translation can be used as the basis for the translation of the help.

The alternative is to translate complete packages module by module (software and help texts combined). This is especially useful for updates and upgrades. In this way the translator can acquire maximum knowledge of the details.

Translate and test module by module. In that way you can divide the work into substantial parts, get a better grasp of the subject matter, are able to consider the terminological differences between modules, and you can assign final responsibility for a module to a single person

1.2.1 Translation subprojects

1 Preparation/planning/glossary composition and validation
2 1st Project: BAAN Software
3 2nd Project: BAAN Help Info

First translate the master modules, because they are part of any system configuration and also relatively easy to translate. Being the foundation and also the smallest package, BAAN Common could serve as a pilot package in order to test the procedure. Of course the sequence of translating packages and modules also depends on commercial priorities.

2. Functional preparation

2.1 Human resources

Ideally, a translation team consists of as many members as there are packages to translate. They are preferably native speakers of the target language and combine knowledge from various disciplines:

- Source language
- Target language
- Field of application (financial accounting, for example)
- BAAN module(s), for example BAAN Finance
- BAAN Translation Tools

Preferably, the team should physically be in one office. Select the members considering their specific skills and know-how. However, because knowledge of the subject matter is as important as linguistic skills are, the translation team must be able to draw support from a team of consultants with expert knowledge of the terminology and the BAAN modules, such as MRP, Purchase, Sales, Shop Floor Control, and so on. In addition, expertise must be available on BAAN Tools and UNIX.

2.2 Product information and training

In preparation of the actual translation work, the translators should:

- Read the document called Software Coding Standards (SCS) for software standards
- Translate and localize the language-specific SCS section called "Text Standards."
- Agree on additional text standards (spelling, punctuation, abbreviations, and so on)
- Read available documentation
- Translate the glossary (Terms and Definitions) provided by Baan
- Study the Baan Translation Manual

Information about new software modules can be obtained from:

- BAAN functions & features (per package)
- BAAN module descriptions (detailed information per package)
- BAAN translation notes (product structure and planning figures)

Projects for the first-time translation of new software should start with product training. It is advisable to invest up to 20% of the total translation lead-time in getting to know the product. This will pay back in terms of quality and time, especially if the same translation team will also translate future package(s) or versions. Product training may consist of:

- Attending BAAN courses
- Working with the software
- Reading available documentation

2.3 Terminology control

Terminology control is one of the most important aspects to assure the quality of a translation project in all phases. Before starting the actual translation, make sure that the terminology is fixed. The BAAN software comes with a Terms and Definitions menu. The central session in that menu is Maintain Terms and Definitions (ttadv7170m000):

An asterisk indicates that there is a definition attached to the term. Mark the record and press the T(ext) button. Definitions are only available in English. If, in spite of the indicator, you cannot find it, change your user language to English.

With each package, Baan can supply a glossary of terms and definitions. Prior to the translation of that software package, study the source terminology and decide on the terms in the target language. Check if official terminology standards are available from a national standardization organization. Decide on the official names for the BAAN packages, modules, and business objects in this stage. Brochures and other commercial documentation that have been translated earlier will already use these names.

Terms must be unique. If possible, try to translate sets of related terms as recognizable sets (for example: order policy/order system/order method).

After translating the glossary, validate it in consultation with application specialists and marketing colleagues. The terminology to be used should be final (as much as possible) at the start of the translation stage. The translated glossary should be assessed and accepted by specialists for the fields involved (consultants/trainers in Manufacturing, Distribution, Finance, and so on). These specialists should have an in-depth knowledge of the terminology used in their field and in BAAN.

Keep the glossary up to date during translation. An up-to-date glossary is also a useful source of information for future translation projects.

On request BAAN will provide the glossary in ASCII format. To import this glossary, proceed as follows:

1 First define a text group with the following name and settings (see the tick):
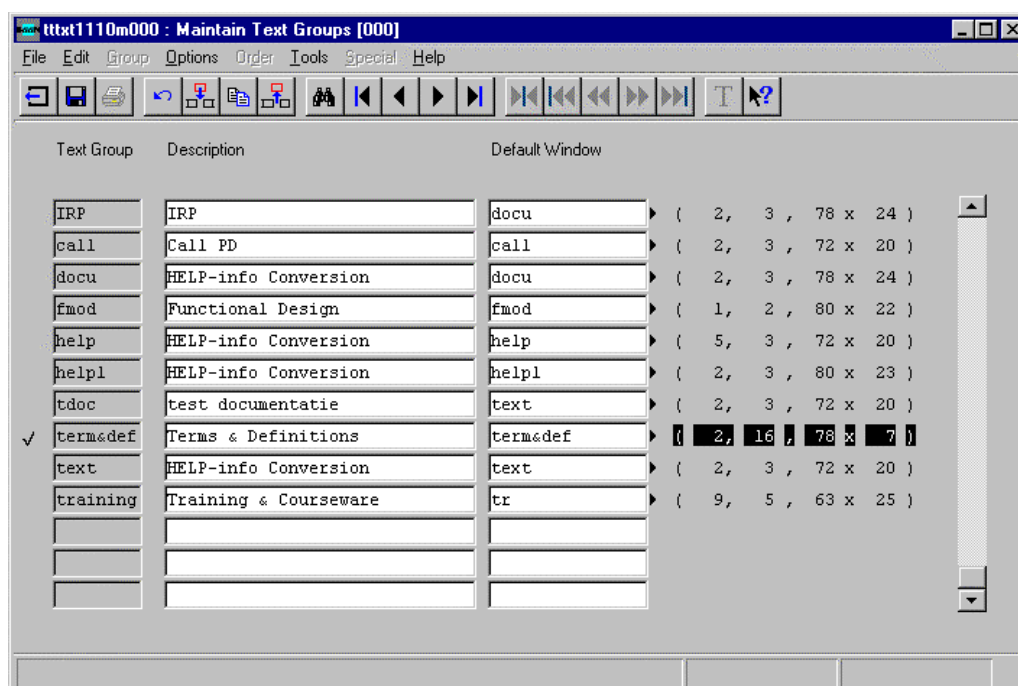
2 Then associate this group with a text window:

3 Then go into Text Management | Text Parameters | Maintenance | Maintain Default Text Group by Text Table Field (tttxt1112m000); it should look as follows:

4 Now you can import the descriptions.

Monitor and evaluate the use of terminology and text standards, particularly at the beginning of the translation phase. This can be done by comparing the work of the individual translators, especially during the first few days. It is often surprising how many (minor) differences occur, particularly when you are working with a team of different people. Checking text standards includes the following aspects:

- Terminology
- Spelling
- Text conventions
- Style
- Punctuation

At this early stage, a high degree of standardization can be achieved with relatively little effort.

3. Technical preparation

3.1 Hardware requirements

You can translate within BAAN or—using the export facility—outside BAAN: on UNIX system or on PCs. Eventually the translated ASCII files will have to be imported into the BAAN translation environment where the translation must be finished and tested.

The following hardware is required:

- A UNIX system.

- For TRITON 2: approximately 350 MB of disk space per language (for software and help).
  For TRITON 3: approximately 750 MB of disk space per language (for software and help).
  For BAAN IV: approximately 600 MB of disk space per language (for software and help).

- Sufficient workstations per translator.

3.2 Software requirements

If you use a PC to access BAAN, you need emulation and file transfer software. Export files (ASCII format) are best translated in a wordprocessor that uses the same character set (ISO8859-1) as UNIX. Most Windows-compatible word processors can handle this.

Baan Development uses Reflection X for emulation, Reflection FTP, for file transfer. For the translation of export files, the vi editor is used in the UNIX environment. On the PCs, Microsoft Word is used.

3.3 Introduction to BAAN Tools

The BAAN system comprises various environments. On the one hand, the software package as the user sees it is known as the run-time or application environment. The compiled application and tools software itself is stored in the run-time data dictionary. The average BAAN user can use the software without having the authorization to change anything.

Developers and translators, on the other hand, must be able to change or translate the software. To do so, you can work in the development environment as a software developer, or —as a translator— in the translation environment, which is a more protected version of the development environment. All types of online help can be translated in the documentation environment, which is also part of the BAAN Tools development environment. The software definitions of the development environment are stored in the development data dictionary (DD).

The separation between the run-time environment and the development environment allows software engineers to further develop the system while the user can continue to work in the runtime environment.

Any changes made in the development environment must be transferred from the development DD to the run-time DD in order to make them visible to the end user. This process is known as Conversion to Run Time. The changes brought about by translation belong to the same category.

3.4 Environment

The BAAN Tools Translation menu allows you to translate all language-dependent software components of an application package. You can translate help texts in the Documentation menu.
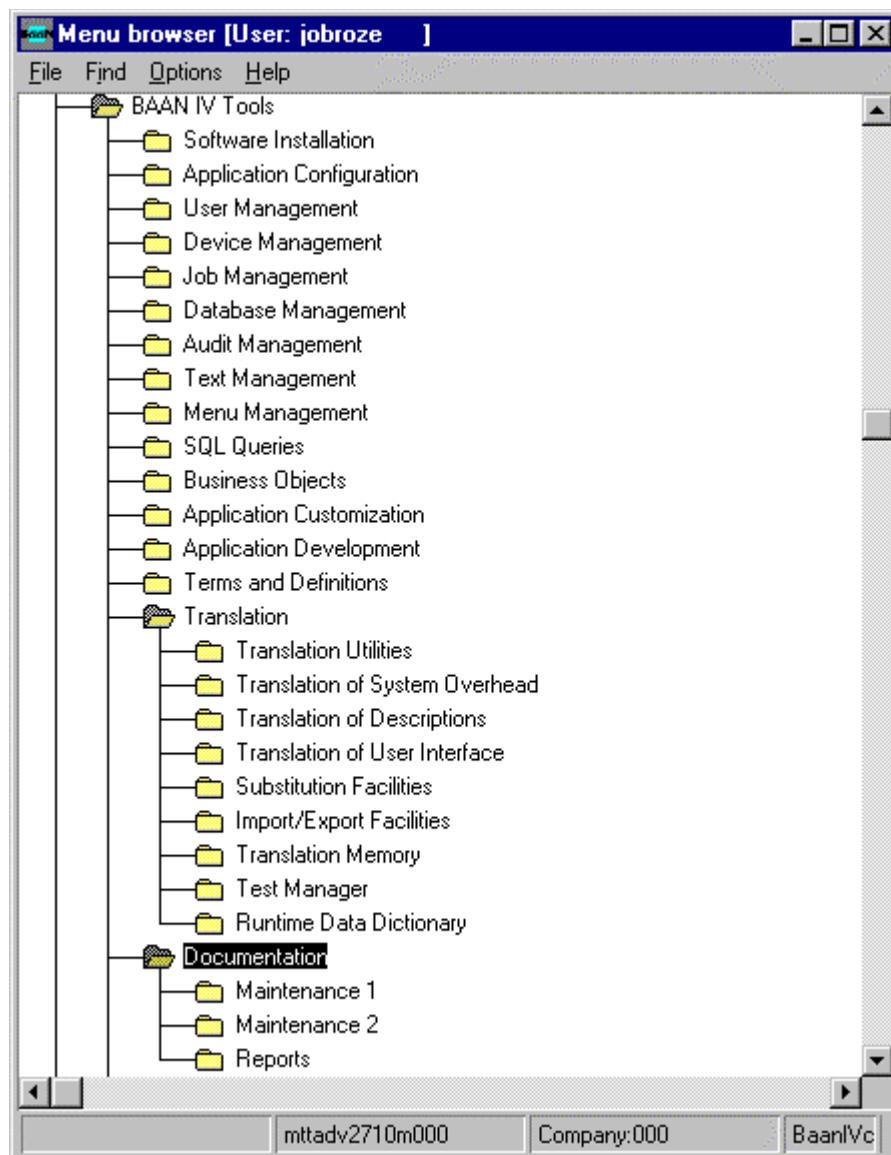
The Translation menu contains all functions relevant to software translation. All descriptions and texts can be edited in a user-friendly translation environment without any risk of damaging the language-independent (technical) data.

You can authorize individual translators to specific sessions in the Translation menu. You can also authorize translators per package, module, session, terminal, and language. It may be useful to assign authorizations for the target language only, to avoid having a translation in the source language by accident.

However, because copying software and help from the source to the target language precedes translation, one person needs authorization for the source language as well, in order to be able to copy. Refer to the responsible BAAN application manager.

Translation Utilities offers utilities for copying, deleting, and testing the software.

Translation of System Overhead contains sessions for the translation of the standard user interface, pull-down menus, status line texts, and so on.

Translation of Descriptions contains sessions for the translation of messages, questions, table descriptions, and so on.

Translation of User Interface contains sessions for the translation of form, report, and menu descriptions.

Substitution Facilities allows you to globally find and replace text strings in the descriptions and help texts for all software components. Its function has now been largely replaced with Export/Import and Translation Memory functionality.

Import/Export Facilities enables you to export software descriptions and help texts so that you can translate or edit them outside the translation environment. This menu also contains a special program to export selective components of BAAN Tools (only the ones used by the average end user).

Translation Memory allows you to store existing translations and reuse them in other software components or in later versions of the same software components.

Test Manager contains sessions to check if labels fit, if all software components have been translated, and if help texts contain valid hypertext codes.

Runtime Data Dictionary contains sessions to activate the translations in the run-time environment.

3.5 Software installation

Before a translation team can start, a system administrator has to take the following steps.

3.5.1 Request software

New software versions can be ordered from Baan by using the standard order form.

3.5.2 Installation

Install the BAAN version to be translated. For major versions, for example, BAAN IV, it is always necessary to do a completely new installation (full update) of the US language version of the software. Then you can use the BAAN translation memory, or whatever other means you have at your disposal, to generate the translation for the new version from scratch.

Minor versions or updates, involving changes and additions, must be installed in an update VRC. To retain the multilevel character of the installation, do not run the Purge Entire Package VRC (ttadv6255m000) session at the end of the installation process.

The standard BAAN Tools system includes all tools and utilities necessary for translation.

3.5.3 Request authorization to translate

Translation involves changing the standard software. As a rule, users are not allowed to do that. That is why you must obtain authorization from a Baan Translation consultant to start translating. This authorization procedure enables the translation tools for the standard software packages. Otherwise you would have to copy all software packages to a custom VRC.

3.5.4 Request language code

Language codes are exclusively issued by the Baan Translation department. They must be unique to ensure several language versions can be distributed and used worldwide, if necessary together in one system.

3.5.5 Define new language code

The target language code and the associated date formats must be defined in the Maintain Languages (ttaad1110m000) session in the BAAN Tools Application Configuration menu.

3.5.6 Package combination

Translate standard software versions under the standard package combination. Localizations and customizations can be translated separately.

3.5.7 Define user data

For every translator who will work on a UNIX system, a system logon code and a BAAN user must be created, for two reasons:

- To be able to work in different VRCs without getting in each other's way
- To be able to trace changes

All BAAN users who will be translating must be defined in the following sessions:

1 User Management | General User Data | Maintain User Data (ttaad2100m000)

2 Developers Data | Maintenance | Maintain General Developer Authorizations (ttadv0142m000) (if authorization should apply for all packages),
   or
   Developers Data | Maintenance | Maintain Developer Authorizations by Package VRC (ttadv0141m000) (if authorization should be set by package and language)

3 Developers Data | Maintenance | Change Current Package VRC of User (ttadv0140m000): (enter any package VRC; the point is that you must specify an initial value)

4 User Management | Convert User Data to Runtime Data Dictionary (ttaad4200m000)

4. Copying software

First the software components must be copied to the target language. There are two types of components to be copied: system overhead and package-specific components. The system overhead is copied once per release or version, because it is valid for all packages. The other components must be considered for each update or release.

4.1 Copy system overhead components

4.1.1 Standard options

Select Copy Standard Options from the Special menu[1] in the Maintain Standard Option Descriptions (ttadv4174m000) session.

---

[1] or press <Ctrl><Shift>[Z]

Fill out the form as shown below, specifying your own language in the target language code field.

4.1.2   Pull-down menus

Select Copy Menu bar/Toolbar Data from the Special menu[2] in the Maintain Pulldown Menu Option Descriptions (ttadv4178m000) session.

----

[2] or press <Ctrl><Shift>[Z]

Fill out the form as shown below, specifying your own language in the target language code field.

4.1.3   Status line texts

Select Copy Status Line Text from the Special menu[3] in the Translate Status Line Texts (ttadv0152m000) session.

---

[3] or press <Ctrl><Shift>[Z]

Fill out the form as shown below, specifying your own language in the target language code field.

4.1.4 Conversion to run-time

Now select the Continue option from the Special menu to activate all options and status line texts using the Convert Standard Options and Status Line Texts to Runtime DD (ttadv0250s000) session. Specify your target language and choose Continue. You must repeat this step after completing the translation.

4.1.5   Tools texts

Select Copy Tools Text from the Special menu[4] in the Translate Tools Texts (ttadv0161m000) session.

Fill out the form as shown below, specifying your own language in the target language code field.

---

[4] or press <Ctrl><Shift>[Z]

Then select the Continue option from the Special menu to convert all tools texts to run-time data dictionary in the Convert Tools Texts to Runtime DD (ttadv0260s000) session. Specify your target language, and the full range of tools text types, then choose Continue.
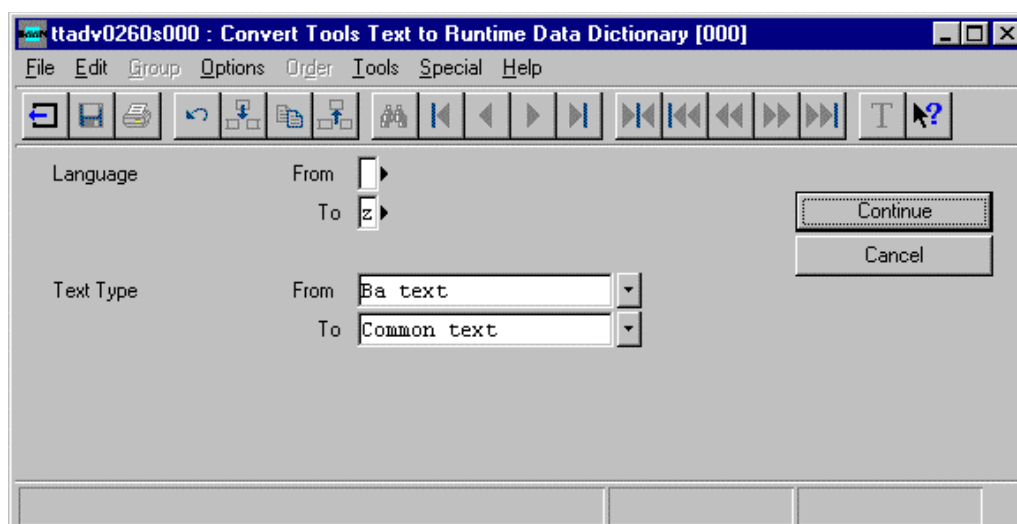
You must repeat this step after completing the translation.

4.2 Copy language-dependent software components

In the Translation Utilities menu, select the Copy/Move Language-Dependent Software Components (ttadv6200m000) session.

Execute this session for each VRC that is part of the current release. Most fields are self-evident. The following three fields are explained in detail.

- Link Forms/Menus/Reports to Source Language
  Forms, menus, and reports are technically defined for US English. If this field is checked, all technical changes made to these elements in the English original will be reflected in the other languages. Not checking this field results in the undesirable situation of having language-specific forms, reports, or menus.

- Copy Variant Label Lengths if Label Exists
  Leave this field unchecked. Otherwise you inster all English label description lengths among the target language labels.

- Label Length Multiplication Factor
  This field is obsolete. It was introduced based on the idea that the average word in German, for example, is 25% longer than in English, but the actual function was never programmed.

You must copy the Tools (tt) package as well. All the Tools software components must be available in the target language, even if only a few will be translated. On the second form, specify the full range for all components.

5. Translating inside the application

5.1 Introduction

In chapter 3 it was explained that programs have been built around all software elements and brought together in a special translation environment to facilitate translation. If you go through all menus and programs one by one, you end up with a fully translated package.

However, this implies that you have to have to enter duplicate translations manually: for each occurrence of the word **customer** you have to enter a translation.

Also, if you want to subcontract (part of) your translation efforts, you have to provide your subcontractor with a full-fledged system.

The current chapter addresses the procedure to translate without leaving the application; chapter 6 deals with the translation outside the system using export files.

5.2 Software translation

5.2.1 Translate system overhead components

To translate the system overhead, call the sessions in the corresponding menu one by one, select the target language and change the English descriptions. Regarding section 4.1.1, Standard options, some additional observations are in place.

There are three elements that can be translated:

1 Help screen option descriptions
2 Option character
3 Help screen

The option character must remain unique! If you plan to change it, consider the following consequences. You would have to adapt (or at least check) the following elements as well:

- Form-specific options
- Text manager options (in Tools)
- Options in DSK, GFD, and GUI
- Form editor options
- Menu editor options
- Report editor options
- Chart editor options
- Enterprise Modeler options

In addition, for the sake of consistency you would have to check ALL references to these option characters in messages, questions, help texts, documents, training material, and, possibly, marketing material!

5.2.2 Translate menus, business objects, sessions, and tables (for a short-term demo version)

First translate and check the highest-level descriptions: those of business objects, menus, tables and sessions. This has the following advantages:

- The menus/business object structure acts as the table of contents of the system.
- Table names describe the entities in the database structure; as such, they constitute the basis of most session and report names.
- Session names are the basis for the descriptions of forms.
- Session names appear frequently in the online help and generated documentation (through hypertext links).
- The translated menu and session titles allow you to demonstrate the outer layer of the package.

The highest-level software components are the textual framework for the individual application packages. They should be translated for each module. For each component the session in which the descriptions can be translated is mentioned.

NOTE

Before you start translating in any of the standard component sessions, you must set the current package VRC. First specify the package and the VRC, then select the Make Package VRC Current option from the Special menu, or press Ctrl+shift+D.

5.2.3 Menus

Menu descriptions are the titles for the individual menus. They are translated in the Maintain Menu Descriptions and Layouts (ttadv3168m000) session, which is available in the Translation of User Interface menu. Some menus contain labels with descriptions such as **Inquiries**, **Maintenance**, **Reports**, or **Miscellaneous**.

5.2.4   Business objects

Business objects are sets of functionally related sessions within BAAN. The same names also appear in the menu structure as modules and submodules. Business objects are a means of structuring the documentation and implementation process. They appear in on-line help and generated user documentation. They are translated in the Maintain Business Object Descriptions (ttadv7101m000) session.
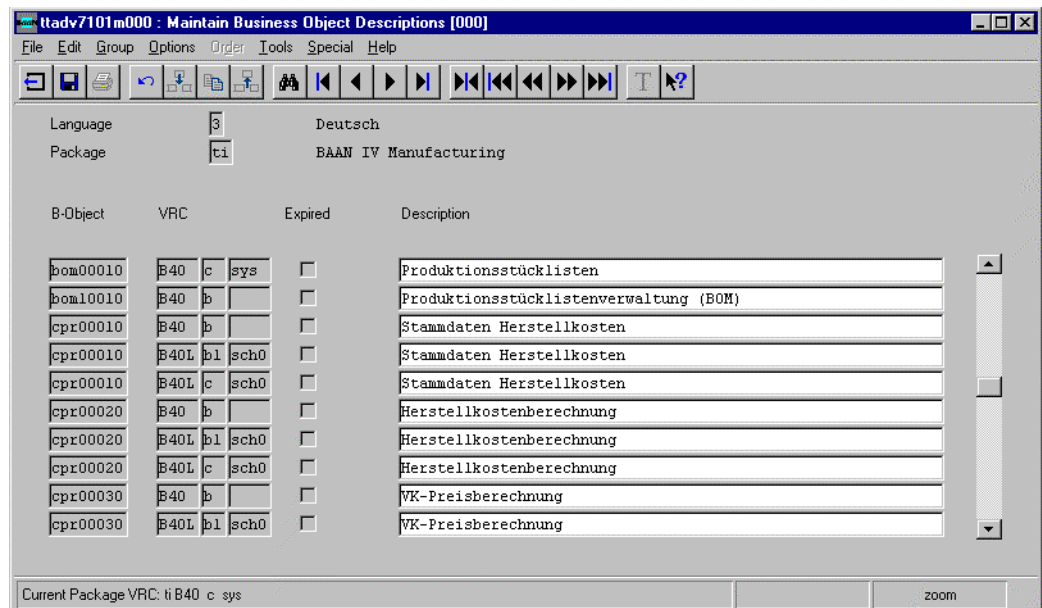
5.2.5   Subfunctions

Subfunctions are a subset of business objects. A subfunction may either be a group of sessions or a group of fields. The purpose of defining subfunctions is to record generic help texts for the sessions or fields that are linked to the subfunction. They appear in on-line help and generated user documentation. They are translated in the Maintain Subfunction Descriptions (ttadv7106m000) session.

5.2.6 Tables

The names of the tables are only visible in the development environment, not to the end user of the software. Table names can be used as hypertext code in help texts. They are translated in the Maintain Language-Dependent Descriptions (ttadv1130m000) session. Set the Description Type to Table.

5.2.7 Sessions

Session descriptions are usually derived from table descriptions, in the sense that a typical session description looks like **Maintain** *table name*. So if you translate the table descriptions first, you have a basis for the session descriptions. There are two types: Session descriptions on form and session descriptions on menu. The former appear as the session names on the forms, the latter are the session titles as they are shown in menus. They may be different, though in the BAAN software both types are always identical. They can be translated in the Maintain Language-Dependent Descriptions (ttadv1130m000) session. You only have to translate, check and correct the first type. Set the Description Type to Session (form).

Afterwards, select the Copy Descriptions command from the Special menu[5] to copy all Sessions on form to Sessions on menu. Check the Overwrite field.

---

[5] or press <Ctrl><Shift>[E]
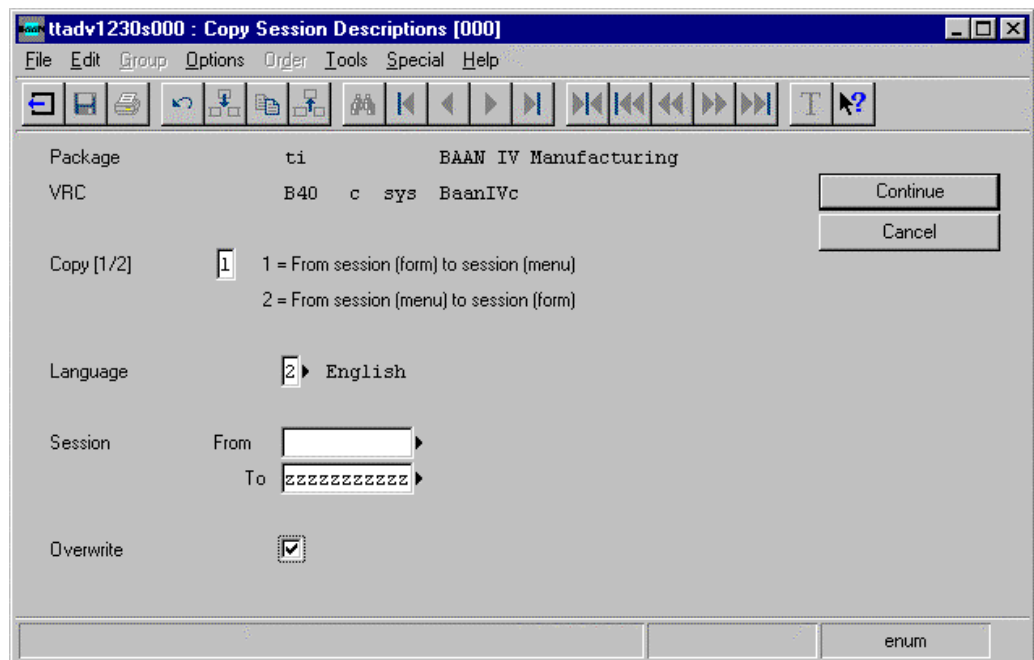
Select the Change Current Package VRC of User command from the Special menu to perform this action for another VRC.

Once you have translated menus and sessions, you can create the menu structure for demo purposes. This is also a good way to check the consistency of the translation.

Create the runtime dictionary for menus (see processing procedure below).

Now you can demonstrate the menus. For a complete demonstration of all screens you have to translate labels and enums as well.

5.3   Translate remaining components

5.3.1   Descriptions

To translate the other language-dependent components, follow the order given in the Translation of Descriptions menu. Apart from session and table descriptions, the Maintain Language-Dependent Descriptions (ttadv1130m000) session is used to translate two other description types: Indices and Charts. Indices can be generated on the basis of translated labels, the chart descriptions are the only thing to be translated at this point.
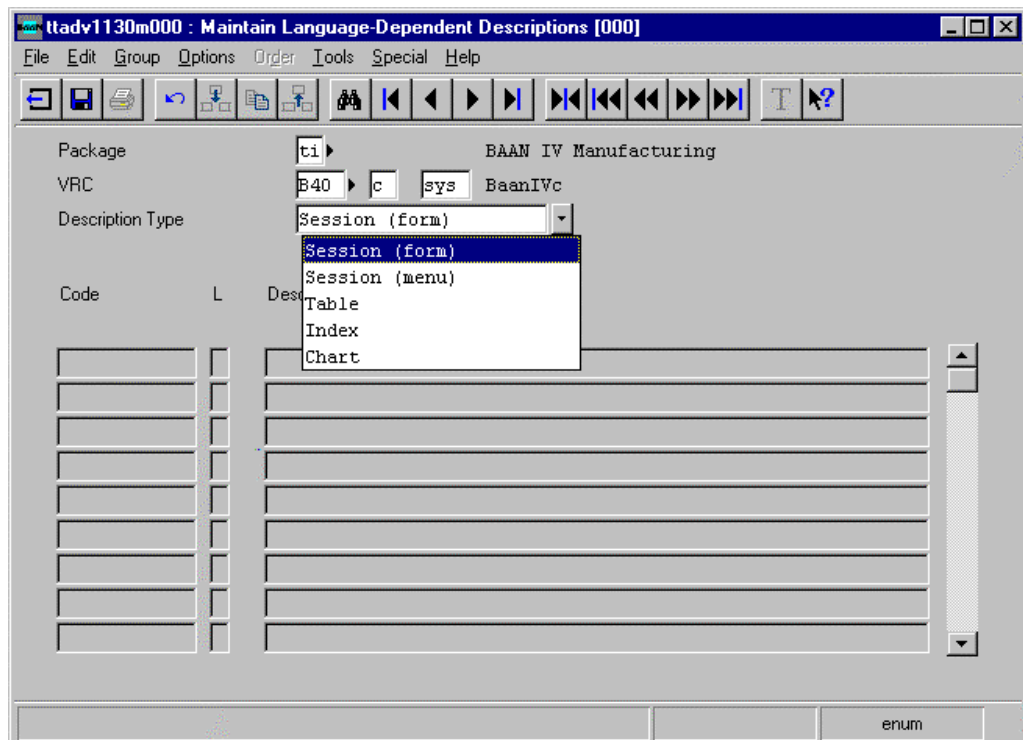
5.3.2 Enums

Enum is short for enumerated listing, a special type of field offering the user a limited choice of predefined options. In the bw user interface these descriptions are presented in a drop-down combo box. They are translated in the Maintain Enum/Set Descriptions (ttadv4102m000) session.

Translate the enum descriptions before the messages and questions, because some of these contain references to enums.

5.3.3 Messages

Messages can be error messages, warnings, pieces of additional information or even variable prompts. They are translated in the Maintain Messages (ttadv4150m000) session.

NOTE:

1 The occasional %s and %d in messages serve as variables for displaying field names or numbers. If one placeholder, for example %s, appears in a message more than once, it is important that the occurrences retain the same sequence!

2 Messages containing <Ctrl> options are shown at the bottom of the screen and are therefore limited to a maximum of 78 positions.

3 Many message codes containing the letter **l** are shown as prompts in forms or reports.

4 Messages may contain an indication for their maximum length, for example: "| max. 10 positions." Obviously the vertical bar (|) and the comment must lie beyond position 10 in the message.

5 The maintenance forms for messages, questions and labels offer an option to search for a text string. Select the Find Pattern in… command from the Special menu[6]. You have to specify if it is upper or lower case. The search starts from the current record onward. To find the next occurrence, select the Find Next (message, question or label) with Pattern option from the Special menu[7]. The search is limited to a single package but will cross VRC boundaries.

There is another session called Maintain Messages (ttadv4151m000), where you can see the message versions for all VRCs. You will find it in Application Development | Messages and Questions.

---

[6] (or press <Ctrl><Shift>[O])
[7] (or press <Ctrl><Shift>[G])

5.3.4 Questions

Questions are messages that require an answer from the user. They are translated in the Maintain Questions (ttadv4166m000) session. They are subject to the same conditions regarding the order of %s, %d, and so on, as messages. See also note 5 on the previous page.

There is another session called Maintain Questions (ttadv4161m000), where you can see the question versions for all VRCs. You will find it in Application Development | Messages and Questions.

5.3.5 Labels

The label file contains the language-specific text elements that are visible in menus, forms, and reports. Labels can also be used in help texts.

There are two sessions in which labels can be translated: Maintain Labels by Package VRC (ttadv1141m000),

and Maintain Labels (ttadv1140m000).

There are three types of labels:

1 Type 1: Generic or term labels
Labels for generic terms are coded as *gen* module components, for example, *tcgen.reports* for Reports or *tigen.machine* for Machine. These labels are used across the modules of a package.

2 Type 2: Table field labels
Labels coded by table field, for example, *tccom010.cust* for Customer.

3 Type 3: Form- or report-specific labels
The most context-sensitive labels are coded by session, for example: *tccoms0100.01* for Print Employee Data by Work Center.

Because they may be linked to various components across modules, it is advised to always copy labels by package, rather than by module, to avoid the overlap.

Compared with the other software components, labels have two extra attributes: a fixed height and length. The absolute maximum is 70 characters long and 3 lines high (1 line of 70, or 2 lines of 35 positions each). Line breaks are indicated by the percent sign. If you want a percent sign displayed in a label, you have to type **%%**.

Menus, forms, reports, charts, tables, and help texts will show the longest label description that fits. The English labels usually have the exact length of the label content. In many other languages the translation will be longer. In order to economize on the number of labels, create one variant that is no longer than the shortest English original. A second variant could be an intermediate abbreviation that would fit most cases. A third variant features the full length, which is necessary when a label is used as a hypertext code in a help text.

If a label description turns out to be incorrect, though you are sure you entered the proper translation, that translation was probably shorter than one in a previous VRC. To avoid this, copy the faulty description to the current VRC and set Expired to Yes.

If you want to add label variants, mark a record and press the Copy button[8]. Set the height and length to 1 and enter the proper description. A message box is displayed, asking if the length must be changed.

It is not possible to add new label codes.

To find out in which menus, forms or reports a label is used, run the Print Where Used Labels (ttadv1445m000) session with the following options:

- Print labels where used
- Print label code on every line

---

[8] (or <Ctrl><Shift>[C])

For a single label this does not take too long, but for a range of labels it is a time-consuming process. In the latter case, it is advised to run this session after working hours and save the result in a file.

See also note 5 in section 5.3.3.

5.4 User interface

To translate the user interface, six sessions are available: for form descriptions, form-specific options, menu descriptions, menu field descriptions, report descriptions, and chart-specific options.

5.4.1 Menu descriptions

The titles for the individual menus. They can be translated using the Maintain Menu Descriptions and Layouts (ttadv3168m000) session. If you first enter the translation, then mark the record and click the Continue button, the menu layout is displayed.

5.4.2 Menu field descriptions

In principle, menu items are session or menu descriptions. It is possible, however, to define other kinds of menu items, for example UNIX shell scripts. The description of a menu item that is not a session or a menu is stored as a menu field description. They can be translated using the Maintain Menu Field Descriptions (ttadv3167m000) session.

5.4.3 Form descriptions

The titles for the individual forms. They are usually the same as the session description, though they may be more specific, for example, if a session has several forms, each with a specific function. Wait until all session descriptions have been translated, tested and corrected. Because form descriptions are usually identical to session descriptions, approximately 95% of them can be copied using the Generate Form Descriptions (Based on Sessions) (ttadv6240m000) session.
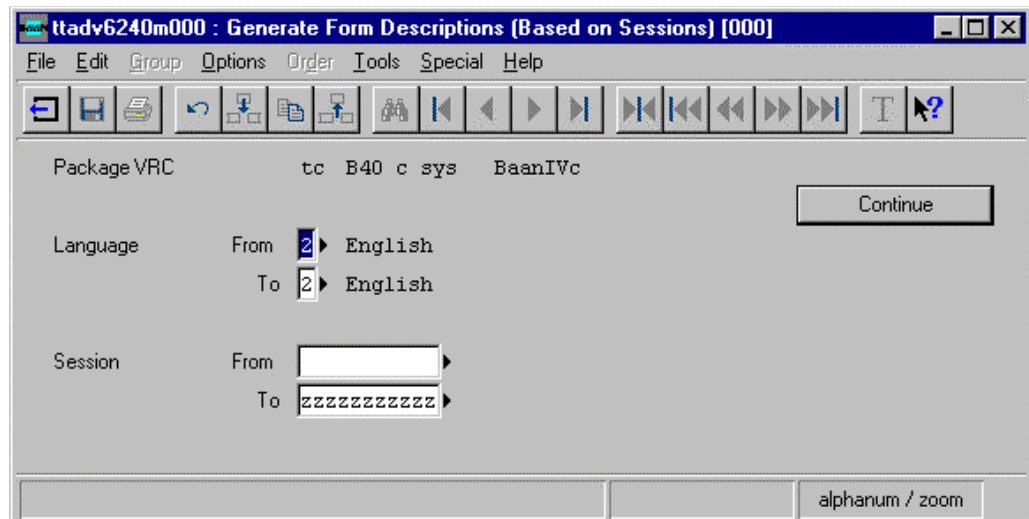
Form descriptions are visible in the HelpViewer and—in the ba user interface—when using the [V] option to switch between forms in sessions with multiple forms.

For the bw interface, this [V] option has been replaced with form tabs. The descriptions in these form tabs are stored as short descriptions of the forms.

Both long and short form descriptions can be translated using the Maintain Form Descriptions and Layouts (ttadv3108m000) session. The short form descriptions are maintained on the second form.

5.4.4 Form option descriptions

In addition to the standard options that function system-wide, a form can have form-specific options. The option description can be translated using the Maintain Form-Specific Option Descriptions (ttadv3112m000) session. The session has two forms: on the first, the full descriptions are translated:

On the second form, the short descriptions are translated. The short form-specific option descriptions are used on buttons.

Because there is much repetition in the option descriptions, it is advised to make a sorted export, translate the options in an ASCII file, and re-import them. See chapter 6, Translating in export files, for more detail.

5.4.5 Report descriptions

As stated earlier, report descriptions are often identical to table names, though variants are possible, depending on the criteria specified in the associated print session. They can be translated using the Maintain Report Descriptions (ttadv3137m000) session.

5.4.6 Chart-specific option descriptions

In BAAN, charts can be fitted with specific options. The associated descriptions are maintained in the Maintain Chart-Specific Option Descriptions (ttchm1121m000) session.

5.5 Software translation dos and don'ts

5.5.1 Do not translate

The following elements need not be translated, either because it is easier (and safer) to generate them, or because they exist for technical reasons only and are no part of the user interface.

- Index descriptions (can be generated).
- Form descriptions (can be generated).
- Session descriptions on menu (can be generated).
- Labels with a letter combination at the end of their code that starts with .**cmb***x*. They refer to combined fields, which is a technical feature.
- Report descriptions from the tffst module that end in **(Generated)**.

The following modules are usually not translated:

- The ISM and PBS modules from the TP package, and the TNR module from the TR package. They were specifically developed for the Dutch market. The software for these modules is available in English; the help texts are only available in Dutch.

- In any package, software elements containing the following module codes: CNV, CON, COR, COV, SQL, TST, ZZZ. They are included for conversion or patch purposes only. End users will not see any of them.

5.5.2 Non-translation/technical changes

In the translation environment you cannot make any technical changes. If you have authorization to change the standard, it is technically possible to make functional and technical modifications to a standard software version in the development environment. However, such changes will not be included in the standard version distributed by Baan Development, because only the translated text elements are added to the English mastertape. It is essential that all language versions of a standard package remain 100% identical. Even the slightest technical change, for example, inserting new label codes or changing label field lengths in forms may cause problems when combining language versions in one customer installation.

During translation, do NOT:

- Add, change, or delete component codes
- Add, change, or delete field data in menus/forms/reports
- Change the length or height of label fields in menus, forms, or reports
- Change the sequence of fields in menus, forms, or reports
- Make any other technical or functional change compared to the English version

Functional modifications due to specific requirements for a language, country, or customer must always be implemented in special localized or customized versions (add-ons). For these purposes special package combinations and VRCs can be defined in consultation with Baan Development.

5.6 Help

The online help information comprises texts on various levels. Basically, every software component can have its own help text. On the basis of the online help the user can generate a user manual per package, module, business object, or session.

5.6.1 Copy language-dependent help texts

Like the software components, help texts must be copied prior to translation. Use the Copy/Move Language-Dependent Text Components (ttadv6220m000) session. Leave the Copy Help based on where used field unchecked if you want to copy the entire range of help text codes.

If you check this field, you can specify the software elements for which you want to copy the help text codes, even down to a single help text.

On the second form you specify the range.

DO NOT COPY:	REASON:
Technical documentation	technical programmers information
Release notes	technical release information
Program functions	not language-dependent
Scripts/libraries	
not language-dependent	

5.6.2 Translate help texts

After copying, change the language of your user ID into the target language, leave the system and log on again. Then go into the Documentation menu, where you can choose to access all help texts through the same session called Maintain Help Text Codes (ttadv1160m000) or access the help files by individual type.

Types of help texts
The following types of help are available (top-down):

TEXT CODE starts with:	TEXT TYPE	TYPE OF INFORMATION
IF	business object help	highest level procedure information
RL	business obj relation help	describes relationships between i-functions
TB	table help	table data definitions, creation and application
MN	menu help	procedure information
SE	session help	objective, definition and how to use the session
FO	form help	more specific than session help
SF	subfunction help	generic description for a range of fields/sessions
EN	domain help	generic description for field input values
FC	table field help	central text for a field appearing in all forms
FI	table field input help	additional text if the field is an input field
FD	table field display help	additional text if the field is a display field
FF	form field help	additional form-specific field text
LB	label help	describes a label (prompt)
MS	message help	describes a message
QT	question help	describes a question

5.6.3 Help text codes

All help texts are independent components. Help texts can be linked to a large number of software components, for example, programs or fields. The VRC of a help text may be different from the VRC of the associated software component.

5.6.4 Hypertext codes

Help texts contain a maximum number of hypertext codes. The meaning of this word in BAAN is somewhat hybrid:

- A hypertext code that does not contain the at sign (@) forms a hypertext link: by double-clicking a highlighted string the associated help is displayed.

- A hypertext code that contains the at sign (@) functions as a label, or placeholder, to ensure that the proper description is displayed.

Hypertext codes start with the percent sign (%). Using them has the following advantages:

- It ensures 100% consistency between software and help text translation
- It reduces the number of translatable lines.
- Any corrections in the software descriptions will automatically be carried through in the help texts.

To find the software term a hypertext code refers to, put the cursor on the code and press <Esc>[A]. The corresponding description is shown at the bottom of the editor screen.
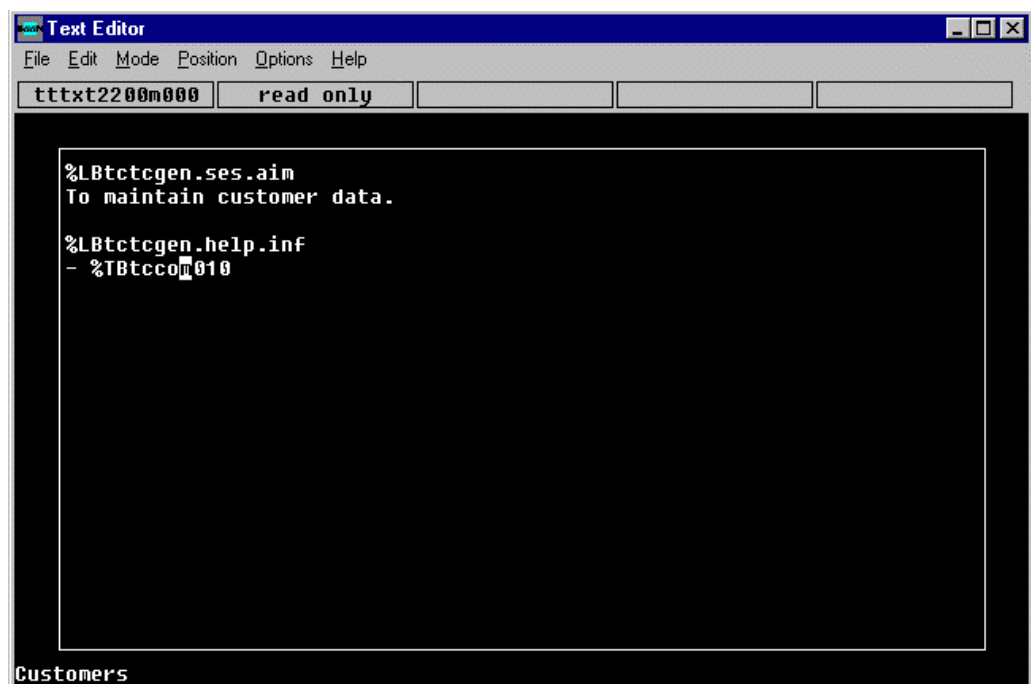
The following types of hypertext codes may occur:

Hypertext Code	Software Component	Displayed As
%IFtccom00010	business object	: Customer Master Data
%SFtccom00010sf0010	subfunction	: How to Sort Addresses
%SEtccom1101m000	session	: Maintain Customer Data
%LBtctccom010.cust	label	: Customer
%LHtctccom010.cust	label + help	: Customer
%TFtccom010.cust	table field	: Customer
%QStccom000001	question	: Do you want to print?
%MStccom000001	message	: Report not found
%FOtccom1101m0001	form	: Maintain Customers
%FFtccom1101m0001cust	form field	: Customer
%TBtccom010	table	: Customer Data
%ENtcyesno.yes	enum	: Yes
%ENtcyesno.no	enum	: No
%OPcont.process	form-specific option	: Continue

Not all strings that start with a percent sign are hypertext codes. There are print control codes (%PFO, %PIT, %PBO, and so on), and there are %GO codes which enable you to jump to another paragraph in the file, marked by a %MK marker. Do not forget to translate the texts associated with such jump facilities.

5.6.5   Comment codes

Some help texts contain comment codes, used to indicate modifications from one version or release to the next. In case of an addition, the codes are placed as follows.

|#SN 160494 (start.new, followed by the date)
new text
new text
new text
|#EN 160494 (end.new, followed by the date).

If a replacement is involved, the codes are placed as follows:

|#SO 160494 (start.old, followed by the date)
|# old text
|# old text
|# old text
|#EO 160494 (end.old, followed by the date).

Then the new text is shown, marked as if it were an addition.

All lines starting with a vertical bar and a number sign (|#) will not be printed or displayed in BAAN. The vertical bar (|) must be on the first position of the line.

6. Translating in export files

The Import/Export Facilities menu enables you to export software descriptions and help texts so that you can translate or edit them outside the translation environment. This menu also contains a special program to export selective components of BAAN Tools (only the ones used by the average end user).

To create an ASCII file, start the Export Language-Dependent Software Components (ttadv6230m000) session. Specify the package, the VRC, the module range, the language, and the output files. If you are going to translate on a PC, you want an MS-DOS-compatible file name. That is why the name format for the output files is:

- For software (Descriptions): s<package>[<module>]<language>.<VRC>, for example: stibom2.40c
- For help texts (Texts): h<package>[<module>]<language>.<VRC>, for example: htibom2.40c

Then check the elements to be exported and press the Continue button.

6.1 Single-level export vs. multilevel export (examples of headers)

If you leave the Incl. Previous VRC field unchecked, only the elements of the specified VRC are exported, resulting in what is known as a single-level export.

If you check the field, the export program considers the elements of all previous VRCs including the specified one. This is called a multilevel export. The difference is reflected in the first two lines of the header of the export file as follows:

Single-level:
2tiB40 c sys
Not also previous VRC

Multilevel:
2tiB40 c sys
True also previous VRC

6.2 Exporting software

6.2.1 Unsorted or sorted

If you leave the Descr. in Unique Sort field unchecked, the exported software descriptions are grouped by element: first all messages, then all questions, and so on, in the order of the elements on the first form. Because many software elements share the same description, in this situation you would have to translate the same string multiple times. This is no problem if you are using some kind of machine translation package, but it is time-consuming if you have to do it all by hand.

If you check the Descr. in Unique Sort field, the descriptions are listed in alphabetical order preceded by the corresponding codes. In that way, you have to translate each description only once. This is more economic, particularly for form option descriptions[9].

---

[9] There is a slight drawback: it is easier to interpret and translate enums in comparison with each other. If they are far apart in the export file because they start with different letters, interpretation becomes more cumbersome. A solution is to create an unsorted export for enums only. The same condition can apply to label variants that start with a different letter than their counterpart(s).

6.2.2   System overhead

To export the system overhead, you have to specify a file for descriptions on the *first* form. Then you can check the elements you want. Export the standard options for reference purposes only; do not translate them.

Example of system overhead export:

```
2tiB40 c sys   DATE: 97-04-15 [15:14]   SORTED
True also previous VRC
#BT 1CU_AREYOUSURE          #132#
Are you sure you want to exit BA (y/n)?
#BT11MR_ARRY_ERR           #132#
Array bound error size %d not in %s
#BT10ATTRNOTLIST           #132#
#BT10DS_ATTRNOTLIST         #132#
Attribute %s not in attribute list\n
#BT10DS_UNKNOWNATTR       #132#
#BT10UNKNOWNATTR          #132#
Attribute of unknown type %d\n
#BT 1CU_ARAISE            #132#
Auto raise
#BT 1CU_BAMESS           #132#
BA MESSAGES
#Og 3 3       #040#
BROWSE
#Og 4 3       #040#
#Og15 2       #040#
#Og16 2       #040#
BROWSE OPTIONS
#BT 1CU_BSHELLMESS         #132#
BSHELL MESSAGES
#BT 2BX_HELLO            #132#
#BT 2HELLO              #132#
BX started\n
```

If a description in this file ends in backslash+n [\n], this character combination must be retained.

6.2.3   Software components

Samples of itm module software exports.

Unsorted, multilevel

```
2tiB40 b1 DATE: 97-04-10 [09:27]
True also previous VRC
#MSitm00045      #132#
Updating item
#QTitm01012      #132#
Delete item data?
```

```
#DMitm.mrpobatch #025#
Batch
#DMitm.mrpolfl   #025#
Lot for Lot
```

```
#TBitm000        #080#
Item Control Parameters
#IDitm0000001    #080#
Sequence Number
#Sfitm0101m000   #060#
Maintain Item Data
#Smitm0101m000   #060#
Maintain Item Data
#LBtiitm.blcm     H: 1
L:19#070#
Inv.Blocked MPS/MRP
#LBtiitm.blcm     H: 1
L:23#070#
Inv.Blocked for MPS/MRP
#FMitm0101m0001   #060#
Maintain Item General/Cost Price
Data
#FOitm0101m0001048#040#
Display Drawing
#Foitm0101m0001048#015#
Drawing
#MDitm000010011   #060#
Item Control
#RDitm020301000   #060#
Delete Items (Error Report)
#IFitm00010       #060#
Item Master Data
#SFitm00010sf0010    #060#
Order Policy
```

Sorted, multilevel

```
2tiB40 b1 DATE: 97-04-15 [12:06]  SORTED
True also previous VRC
#LBtiitm001.abcc    H: 1 L: 8#070#
ABC Code
#IFitm00040      #060#
#MDitm004010011  #060#
#RDitm140101000  #060#
#TBitm011        #080#
Alternative Items
```

```
#DMitm.mrpobatch #025#
Batch
#DMitm.umerbreak #025#
Break
```

The numbers between hashes (#) indicate the maximum number of characters for a software element. Labels have two additional attributes: a length and a height indicator. The exact meaning of abbreviations such as LB, RD, is available in the 40-line header of each export file.

MS: Message, max 132 characters
QT: Question, max 132
DM: Enum Description, max 25
TB: Table Description, max 60
ID: Index Description, generated, max 60
Sf: Session Description on Form, max 60, use standard verbs!
Sm: Session Description on Menu, max 60 (Sm = Sf description)
FM: Form Description, generated
FD: Form Field Description, not visible; never translated
MD: Menu Description, max 60
Mf: Menu Field Desciption, max 60
RD: Report Description, max 60
LB: Label Description, variable length, absolute max = 70
FO: Form Option Description, max 40
Fo: Form Option Short Description, max 15
CH: Chart Description, max 60
IF: Implementation Function Description, max 60
SF: Subfunction Description, max 60

6.2.4 Exporting labels

With the exception of the system overhead, all software elements can be exported by module, because of their module-specific code. However, there are quite a number of labels that are coded generically, for example tdtcgen.date. As a consequence, the recommended procedure if you are translating labels, is as follows:

1 Export the labels by module.
2 Translate the export files.
3 Import the translated labels.
4 Export the labels for the entire package.
5 Translate the descriptions that are still in English.
6 Re-import all labels.

Of course you can also export and translate the labels in one single run per package.

6.2.5 Most economic export

In order to spend the least effort on translating a software export file, set the check boxes on the first *and* the second form as shown below:

You do not export Index descriptions because you generate them after translation, and you do not export Form Field descriptions because they are not used anymore.

6.3 Translating software in an export file

When you are translating the exported software files, pay attention to the following guidelines, apart from the ones mentioned in section 5.5, Software translation dos and don'ts.

- You do not need to change label length indicators.
- You may add extra records (code and description) for label variants.
- Do not translate EXPIRED labels. That indicator is there for reference only.
- Do not alter the very first two lines of the export file.
- Be very cautious with find and replace: you could damage component codes.

6.4 Exporting help texts

Help texts are also exported using the Export Language-Dependent Software Components (ttadv6230m000) session. Enter a File for Texts.

Most help texts are coded by module. An exception to this are a number of business object help texts, subfunction help texts and label help texts. This implies that it is advisable to follow a procedure similar to the one for labels in section 6.2, Exporting software.

6.4.1 Sequential or logical order

If you leave the Help Texts in Logical Order field cleared, the help texts are exported according to their type: first all codes that start with FC, then those starting with FD, and so on. If you check this field, the export file shows all table help and field help together with the associated session help, so that you have a clearer idea of the relationship between the texts.

6.4.2 Types of help texts

The following types of help are available:

Text Code	Text Type	Type of Information
IF	business object help	highest level procedure information
RL	business object relation help	describes relationships between i-functions
TB	table help	table data definitions, creation and application
MN	menu help	procedure information
SE	session help	objective, definition and how to use the session
FO	form help	more specific than session help
SF	subfunction help	generic description for a range of fields/sessions
EN	domain help	generic description for field input values
FC	table field help	central text for a field appearing in all forms
FI	table field input help	additional text if the above field is an input field
FD	table field display help	additional text if the above field is a display field
FF	form field help	additional form-specific field text
LB	label help	describes a label (prompt)
MS	message help	describes a message
QT	question help	describes a question
RD	report help	describes a report

6.4.3 INFO codes

On the second form you can specify what kind of text you want to export. If you check the Hypertext Info Lines field, the descriptions corresponding with each hypertext code in a help text are appended to that help text in the export file. This provides the translator with the required context information.

Example:

```
#H helptextc  SEcom1101m000
%LBtctcgen.ses.aim
To define and maintain the data about customers and prospects.

%LBtctcgen.ses.use
The data defined in this session can be used in the module "%IFtdsls10010".
After you have entered the data in the field "%TFtccom010.cuno", the system will come up with
the message "%QStccom00001".

#INFO: %LBtctcgen.ses.aim = SESSION AIM
#INFO: %LBtctcgen.ses.use = SESSION USE
#INFO: %IFtdsls10010 = Sales
#INFO: %TFtccom010.cuno = Customer
#INFO: %QStccom00001 = Actual customer or prospect?
```
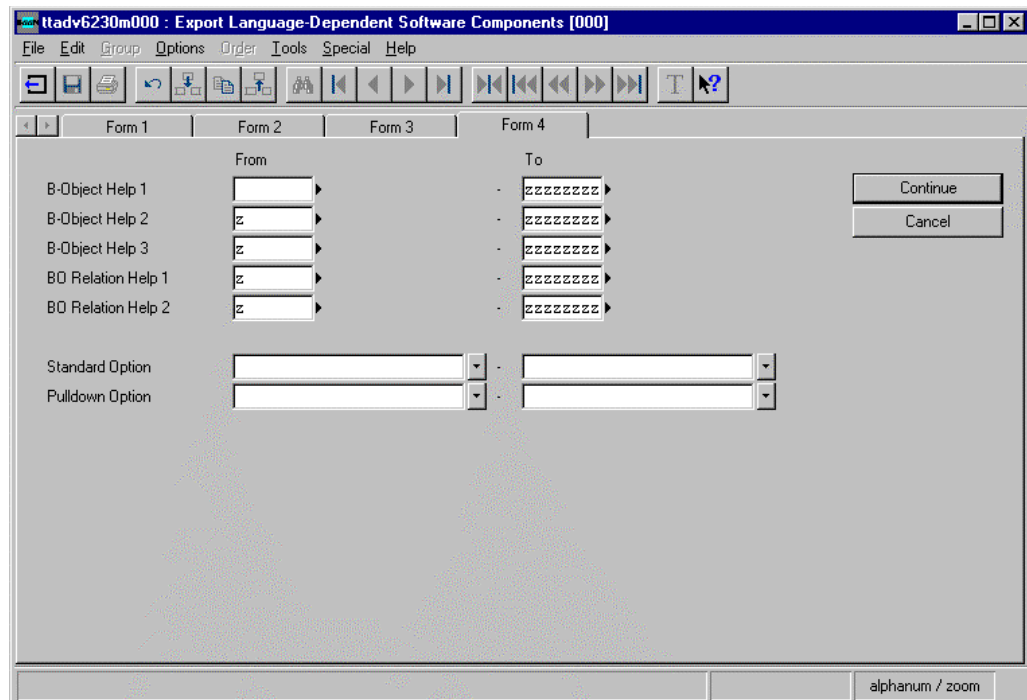
On the third form, you can specify the ranges for the elements you checked on the first from. On the fourth form, only specify the first range; for the other software elements no help texts are available.

6.5 Translating help texts in an export file

- If you translate on a PC, use an editor that can handle the ISO8859 character sets.

- Do NOT use a DOS-based word processor (DOS uses a different character set: you would need a conversion program to convert accent characters and graphical lines)

- Use a non-proportional font (otherwise you cannot set the max. of 70 characters per line).

- Disable the option to avoid two initial capital letters, otherwise hypertext codes will not be recognized (for example, "%SE…" would become "%Se…")

- There are quite a few help texts that contain boxes, or frames. In a Windows-compatible word processor these boxes are displayed as a combination of special characters for the lines and the corners. You can add, move, or remove those characters as required. If you save the file as specified below, they will be imported properly.

- Set the document margins in such a way that there are max. 70 chars per line (the absolute maximum to import is 72). Create a template document with these settings and use this to process future help files.

- After finalizing the translation, save the file as **Text only with Line Breaks** (Word option; other editors may phrase this differently). During the translation process you may save as the file as the editor's standard file.

6.6 Importing

After everything has been translated, you can start importing the data back into the system. The Import Language-Dependent Software Components (ttadv6231m000) session is used for both software and help texts.

6.6.1 Importing software

Before running the import program, check the ASCII file. All codes must start with the number symbol (#) on the first position of the row. Also make sure that no codes have been damaged. This is especially important if the file has undergone technical actions or has been transferred between systems or formats.
Refer to Appendix B, Pre-Import Checklist.

The import program reads the ASCII file, looks for the component code in the system and then overwrites its description. All components must exist in the target language before they can be imported. The import process does not create any new components.

With labels this is different, because label lengths can vary from one language to another. When the import program finds a label code, it first deletes all identical label codes, then adds the codes from the import file. In other words, for each set of label codes in the export file the full set of identical label codes in the database is rewritten.

6.6.2 Importing help texts

For texts, the import session only considers the range of help text codes, not the range of component codes. Specify the import file on the first form.

On the third form, check the Help Text Codes field. The Overwrite Text which Has Been Updated after Exp. field was introduced for the following reason. While help text are being translated in an export file, you are not supposed to make any changes to them in BAAN. If you check this field, the changes in BAAN will be overwritten. If you want to retain those changes, you have to clear this field.

On the fourth form, you can specify the help text codes (usually the full range):

6.6.3   Error messages and handling

Errors occurring during the import process should be sent to a printer or to a file (device ASCIF). Use speaking file names based on package/module code and language code, for example, swtc4.imp. Possible import errors are:

For software:

1 Not found
This means there is a mismatch between a component code in the ASCII file and its counterpart in the system, either because the code in the file has been damaged, or because it has been deleted or recoded in the system.

2 The descriptions/texts of the following records are too long
This means the component description exceeded the maximum length. The description has been imported, but truncated. Shorten the description in the export file and import it again, or rephrase the description in the application.

For help texts:

3 Help Updated After Export
This means the text was changed after it was exported for translation. Because there is only one date for all texts, whether they are in Dutch, English, German or French, this warning does not necessarily mean that the, English, source text was changed.

4 Word wrap in text
The maximum line length for help texts is 72 positions. If a line in an export file is longer, the import program will wrap it to the next line in the help text file and specify the text in this message. This may also apply for vertical lines that are part of a box, where the rightmost one is wrapped to the next line.

7.   Verification

Software verification ensures that the product is in accordance with technical standards. In other words: the software translation must be complete and technically correct. The technical standards are defined in the Software Coding Standards (SCS) document.
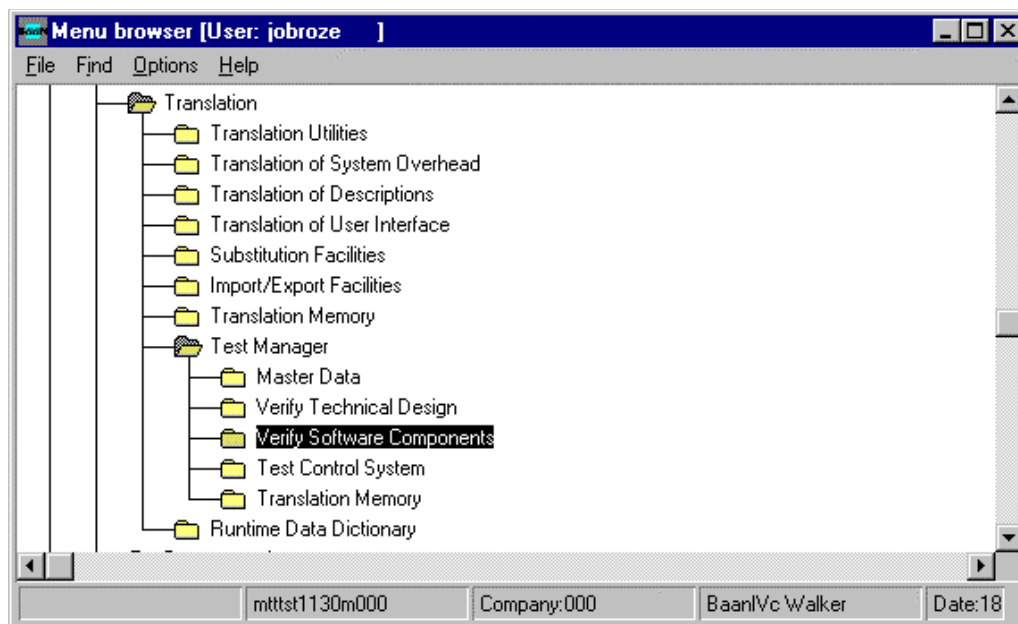
7.1 Verifying software components

The Verify Software Components by Language (tttst1430m000) session checks if all components are available in the target language and have been translated. You can run this session for a single VRC or for the entire VRC tree.
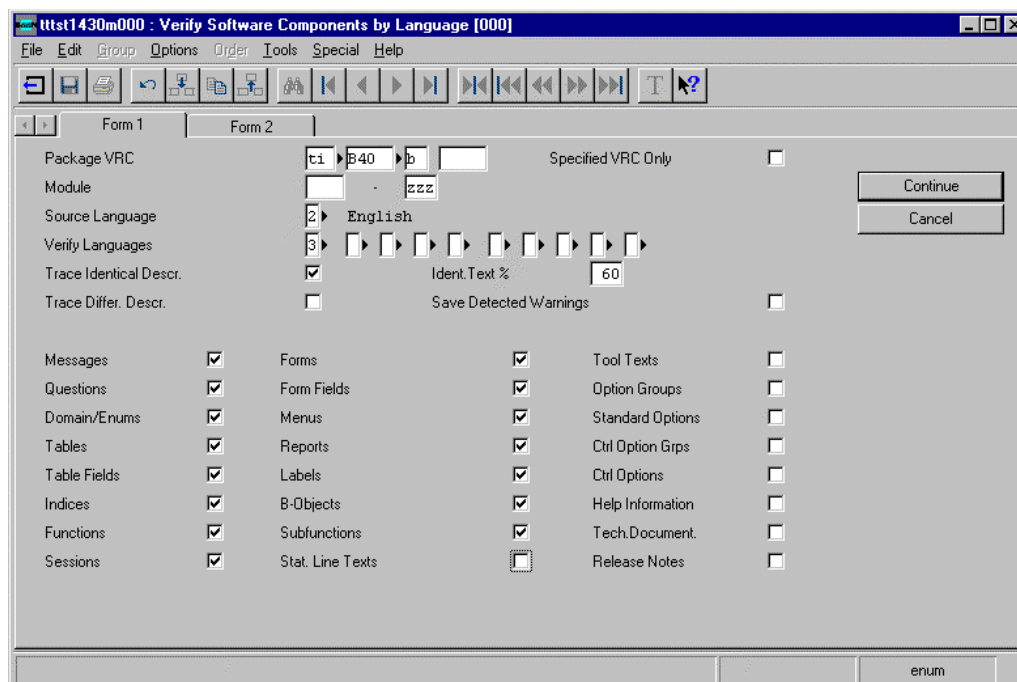
The test programs must be run for each package. Press Continue[10] to start the session or select the Make Job option from the Tools menu[11] to define a job[12]. If components have not (yet) been translated (for example, form descriptions), this session may list a lot of identical descriptions. That is why it is recommended to send the output to a file, not to a printer.

The output contains errors of the following types:

- **Not found**
  Solution: Run the Copy Language-Dependent Software Components (ttadv6200m000) again, setting Overwrite to No; then translate.

- **Identical description**
  Solution: Translate (if necessary: the word Name, for example, is identical in English and German).

After all errors have been handled, the program can be repeated with the Save Detected Warnings field checked. All remaining warnings are then stored in the test

---

[10] or enter **y** in the Choice field

[11] or enter **g** in the Choice field

[12] A job allows you to run a number of sessions in batch. Jobs can be executed after office hours and can be repeated at any time.

database. Many identical descriptions will be acceptable, for example, for all .**cmb***x* fields, or words such as MRP, Time Fence or Code. If you check the Accepted field in the Maintain Verification List by Software Component (tttst1110m000) session, such cases will not be reported the next time this program is run.

7.2 Handling label length

7.2.1 Example

As stated earlier, the standard program displays the longest label that fits. To determine if a label fits, it matches its technical length with the length of the label field on the form, report or menu.

Suppose there is only one label variant, whose description (12 positions) is shorter than the technical length (20 positions). The field on the form where that label is used, is 17 positions. Then the text will not be displayed, even though it is short enough to fit: the technical length is too large.

7.2.2 Label stripping

The Strip Labels (ttadv1250m000) session (see the Translation Utilities menu) optimizes the label file by redefining the technical label length based on the actual label description. In the example, it will cut down the label length from 20 to 12. Any duplicate labels that may occur in the process will be deleted. You must run this program for each individual VRC in process.
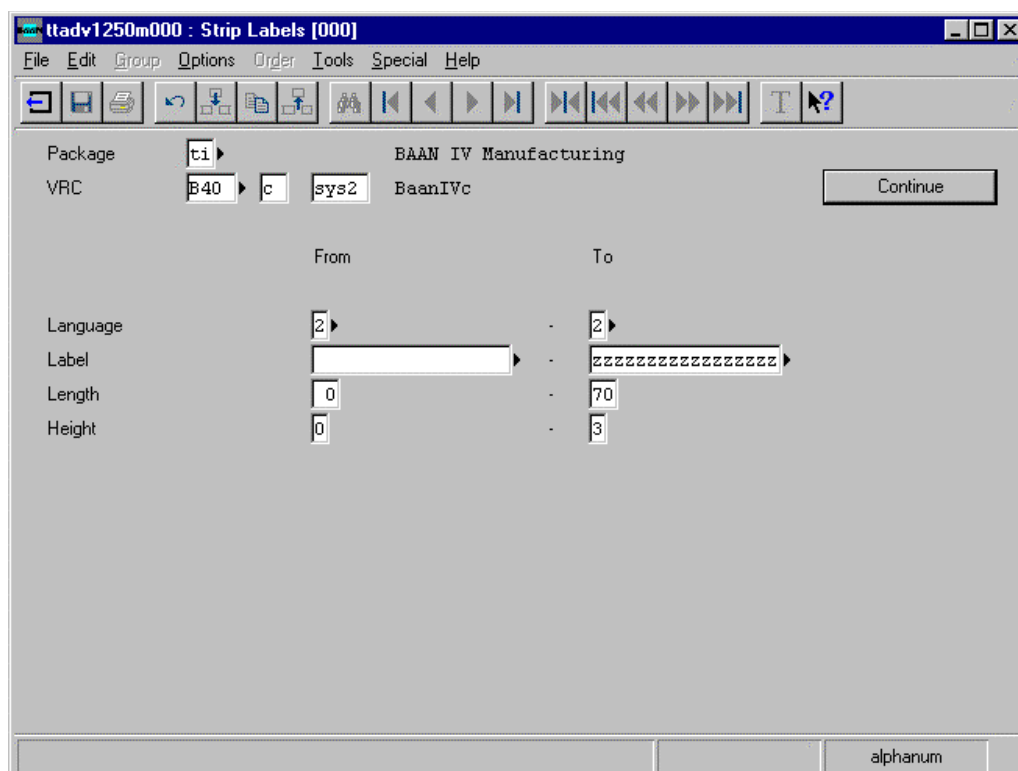
7.2.3 Verify labels

The Verify Labels by Software Component (tttst1470m000) session checks if each label field in menus, forms and reports has a label with the correct length that fits. Run the Strip Labels (ttadv1250m000) session first to reduce the number of errors.

Always specify the highest VRC only!

This session automatically goes through all VRCs from the highest one down (multi-level). If you specify a range of VRCs, it will first do the lowest VRC, then the second followed by the lowest, then the third, the second and the lowest again, and so on.

Select the Check Labels by Label Code report to get the most concise report.

Then select the output type: display, file, or printer. The report looks as follows:

The first line starts with tf B50 a fd01. That is the package VRC where the faulty label occurs. The code of that label is tfacp.print.fully. Its description in language 3 is longer than the available maximum of 20 positions in form acr2420m000. You must abbreviate the label, by copying an existing label variant to one with the required height and length using the CTRL+C option in the Maintain Labels (ttadv1140m000) session. Of course, this action automatically solves the error in the second line as well.

The Label column can also contain the string Not found. Then you must execute the Copy Language-Dependent Software Components (ttadv6200m000) session again for that label; clear the Overwrite field.

7.3 Checking menus, forms, and reports

The translated descriptions are immediately visible in the Translation Environment. The main part of the testing is to view all menus, forms, and reports in order to check the labels in their proper context.

The text elements in menus, forms, and reports are stored in the label file. You can view the layouts of menus, forms, and reports in the relevant sessions in the Translation of User Interface menu. The labels can be corrected by zooming to the label file.

The sessions for menus, forms, and layouts function in the following way:

1 Press CTRL+F to select language, VRC, and code.

2 Press CTRL+D to make the VRC current.

3 Press C[13] to edit the title.

4 Select the line number and press Y to call up the layout[14].

5 In a layout, press CTRL+N or CTRL+P to move the cursor[15].

6 On a label field, press CTRL+Y[16].

7 Press C for change.

8 Press E to exit the label file.

9 Press CTRL+A to leave the form (quit without saving; the text changes have been saved in the label file).

For more context information about the labels in a form you can call the form field attributes by pressing CTRL+F[17] in a data field. If the Help field is set to Yes, you can view the form field help by pressing H. In the translation environment this information is read-only. Quit the help text by pressing ESC, Q.

Make sure that corrections are carried through consistently across component types, modules and packages.

Keep a record of general and terminology changes, as well as of the forms and reports that have been tested.

---

[13] or place the cursor in the field

[14] or mark the record, then double-click it

[15] or use the arrow keys, or the mouse

[16] or double-click it

[17] or double-clicking

You can also print the form layouts, using the Print Forms (ttadv3400m000) session in Application Development | Forms. Sort the forms by language, package, form, and VRC (option 2), and check the Print Layout only field.

Click Continue and select Forms (Field Data).

Click OK. Then specify if you want a file or a printout.

If you specify a file, create and run the following UNIX script to eliminate redundant and empty lines.

```
clear
echo "Enter name of file to be stripped: \c" ; read FRM
sleep 2
echo "Creating ${FRM}.g"
grep -v ^$          ${FRM}   > ${FRM}.a
grep -v ^Dat        ${FRM}.a > ${FRM}.b
grep -v ^TRITON     ${FRM}.b > ${FRM}.c
grep -v ^------     ${FRM}.c > ${FRM}.d
grep -v ^Language   ${FRM}.d > ${FRM}.e
grep -v ^Taal       ${FRM}.e > ${FRM}.f
grep -v ^LAYOUT     ${FRM}.f > ${FRM}.g
rm ${FRM}.[a-f]
```

7.4 Verifying help texts

Before converting them, make sure all help texts have been translated and are also technically acceptable.

7.4.1 Checking hypertext codes

During translation, hypertext codes may have been changed inadvertently, for example due to the Word option to avoid double capitals at the beginning of a word (all hypertext codes start with double capitals: %TBacp500), or due to a too rigorous find/replace action. Use the Verify Technical Design (tttst1400m000) session to check if the hypertext codes in the help texts refer to existing software components.

You can store all settings under a predefined Verification Code, or use the Save Verification Defaults and Get Verification Defaults options from the Special menu.

Select all components that have help texts; on the forms 2 - 12 check the Help Information and Hypertext fields, for example:

Send the output to a file. Only the lines containing the word **help** are relevant for translation. You can disregard the errors about enum codes that contain more than 1 dot (.), for example ENtfgld.trans.com.

Exceptions:

Acceptable warnings, words that are identical in two languages (Code, Project, MRP, and so on) can be stored using the Maintain Verification List by Software Component (tttst1110m000) session.

7.4.2 Completeness check

To check if all help texts have been translated, run the following UNIX script in the directory where the import files are located.

```
1     for i in $*
2     do
3     echo testing $i...
4     echo "UNTRANSLATED LINES NOT ALLOWED:" > $i.error
5     grep -ine "^Th " $i >> $i.error
6     grep -ine " the " $i >> $i.error
7     grep -ine " for " $i >> $i.error
8     grep -ine " a " $i >> $i.error
9     grep -ine " with " $i >> $i.error
10    echo "                    " >> $i.error
11    echo "COMMENT CODES SO AND EO NOT ALLOWED:" >> $i.error
12    grep -ine "|#" $i | grep "SO" >> $i.error
13    grep -ine "|#" $i | grep "EO" >> $i.error
14    echo "                    " >> $i.error
15    echo "SPACE IN CODE LINES NOT ALLOWED:" >> $i.error
16    grep -ine " #H" $i >> $i.error
17    grep -ine " |#" $i >> $i.error
18    echo "                    " >> $i.error
19    echo "REMAINING REFERENCES TO GEM-DRAWINGS:" >> $i.error
20    grep -ine "\.gem" $i >> $i.error
21    echo "                    " >> $i.error
22    echo "REMAINING REFERENCES TO TRITON or BAAN:" >> $i.error
23    grep -ine "triton" $i >> $i.error
24    grep -ine "baan" $i >> $i.error
25    echo "                    " >> $i.error
26    echo "LINES LONGER THAN 72 CHARACTERS:" >> $i.error
27    cat $i | cut -c72-150 | sort -u >> $i.error
28    done
```

The output file is called *export file*.error. For each error it shows the number of the line in the file. You can correct those errors in the export file, then import it. See Appendix A, Unix and vi, about how to process a file using vi.

7.4.3 Additional checks

Check if the boxes have been imported properly.

8. Translating processing

Up to this point, activities have taken place in the development environment. To make the translations available in the run-time environment, additional actions are required. There are six different actions to be taken before the end user can use BAAN in his own language.

- System overhead must be converted.
- Tools texts must be converted.
- Sessions, tables and enums must be converted.
- Forms and menus must be converted.
- Reports must be compiled.
- Run-time help must be created.

8.1 Create run-time data dictionary

8.1.1 System overhead

In any of the first five sessions of the Translate System Overhead menu, select the Continue option from the Special menu to activate all options and status line texts using the Convert Standard Options and Status Line Texts to Runtime DD (ttadv0250s000) session. Specify your target language and choose Continue. See chapter 4, Copying software, for the corresponding form layouts.

8.1.2 Tools texts

Then select the Continue option from the Special menu to convert all tools texts to run-time data dictionary in the Convert Tools Texts to Runtime DD (ttadv0260s000) session. Specify your target language, and the full range of tools text types, then choose Continue. See chapter 4, Copying software, for the corresponding form layouts.

8.1.3 Sessions, tables, and enums

First of all, the session, table, and enum descriptions must be made available in run time. To that end, run the Create Runtime Data Dictionary (ttadv5210m000) session in the Runtime Data Dictionary menu. Because of the table reconfiguration this is a time-consuming process, so preferably run this session after working hours. Fill out the proper package combination and the rest of the fields as specified below.

NOTE

You must run this session for the BAAN Tools (tt) package as well.

Run this session if you have used sessions such as:

- Copy Software Components to New Package VRC (ttadv6265m000)
- Delete Software Components by Package VRC (ttadv6250m000)
- Copy/Move Language-Dependent Software Components (ttadv6200m000)
- Import Language-Dependent Software Components (ttadv6231m000)
- Maintain Enum Descriptions (ttadv4102m000)

8.1.4 Forms and menus

Secondly, the changes in forms and menus (more specifically: the label translations) must be made available in run time. To that end, run the Create Runtime Data Dictionary (Forms and Menus) (ttadv5210m000) session in the Runtime Data Dictionary menu. Set the Dump field to 2.

You must run this session for the BAAN Tools (tt) package as well.

8.1.5 Tools

Run the Create Runtime Data Dictionary for Tools (ttadv5213m000) session to make the translation of BAAN Tools components available at runtime.

8.2 Compile reports

Run the Compile Reports (ttadv3240m000) session in the Translation Utilities menu to make the changes in reports visible at runtime. This session must be run for every package VRC. Specify the range and the other settings as shown below.

NOTE: You must run this session for the BAAN Tools (tt) package as well.

8.3 Create runtime help

To update the run-time data dictionary for help texts, run the Create Runtime Help (ttadv5230m000) session for each package.

NOTE

You must run this session for the BAAN Tools (tt) package as well.

9. Validation

9.1 Software

Software validation ensures that the product is in accordance with functional specifications, meaning that the translation must be adequate and functionally correct.

In practice, each software module should be tested at run time by a second member of the translation team and an expert in the relevant field of application (consultant or trainer). You may even want to involve a customer who is willing to act as a pilot.

9.1.1 Runtime test

After the technical processing procedure the translation will be visible in the application environment. Depending on the language code in the user data, you can start up the translated language version.

Final testing
1 Test package at run time: 2nd translator

2 Test package at run time: package expert

Final corrections
Corrections can be made in the translation environment or, if terminology is involved, in an export file.

Repeat the technical processing sessions to make the changes available in the application environment.

9.2 Documentation

Another option to check the screen layouts on paper is to generate documentation. If, at the start of the translation project, you copied all software components, including help texts, you can check the translated forms and read the help information. If you print the documentation in the source and the target language, you can perform an even more thorough check.

9.2.1 Generate user documentation

In order to print user documentation, proceed as follows:

- Go to Documentation | Reports
- Start the Print User Documentation (ttadv7450m000) session

You can generate user documentation by menu, business object, or session. All components associated with (a range of) menus, business objects, or sessions will be printed.

On the second form you can specify the types of help you want to print. For example, to print just the form layouts and associated help information, check the Form Help and Form Layout fields. The setting on the form below is the standard for validation purposes.

On the third form you can specify if you want the help text codes to be printed as well, and if you want to skip identical texts (advisable).

The fourth form is not used for printing for validation purposes.

10. Translation maintenance/upgrade

10.1 Translation memory

A translation memory (TM) is a database of language-dependent descriptions, where you can store existing translations to reuse them for other component descriptions or in later versions of the same components. The current BAAN IV TM supports descriptions (software) except labels.

10.1.1 How to upgrade a translation using TM

Example: an upgrade from B40 a1 to B40 b.

Conditions:
- The existing translation of the relevant modules must be complete and correct.
- The B40 b upgrade must be installed as an update installation on top of B40 a.

Procedure Steps:
1 Set your company to **000**

2 Write all translated descriptions to TM by running the Update Language-Dependent Descriptions in TM (tttst6211m000) and the Update Language-Dependent Texts in TM (tttst6212m000) sessions. Specify the latest translated VRC (B40 a1) and check the Incl. Previous VRCs field.
Although the translation of labels cannot be generated, it must be possible to use label descriptions for the generation of other translations, so the update must include labels.

3 Copy the B40 b software and help from language code 2 to the target language in the Copy/Move Language-Dependent Software Components (ttadv6200m000) session.

4 Run the Translate Software based on TM (tttst6200m000) session.

This session can only be run for a single package VRC at a time, but you can do multiple languages with it. It also generates a log file (called errorfile on the form) and stores it in the path you specify.

For software there are three different log messages:

– **MStiitm0203aa not found in exportfile**
This means that the component is new. Translate it manually.

– **MStiitms0024 description differs**
This means that the description of the component in the source language was changed since the last time it was translated. Check it against the English original and adapt the translation if necessary.

– **Sftiitm0113m000 translation generated based on FMtdilc0110m0001 by other package**
This means that the description for this component has been translated with the aid of an existing equivalent in another package. Check it against the English original and adapt the translation if necessary.

The log file ends in four lines of statistics:

– **Updated/Inserted records by equal description : 1633**
– **Updated/Inserted records by almost equal description: 13**
– **Updated/Inserted records by generated description : 128**
– **Not Updated/Translated records : 621**

5 Recopy labels (for all packages, setting Overwrite to No)
6 Check the lines of the B and C categories in the TM log file.
7 Translate the remaining components manually (this includes all labels, because these are not supported by TM).
8 Run the verification programs.

If you want to see if all translations have been stored properly, you can view them in the Maintain Translation Memory of Descriptions (tttst6100m000) session.
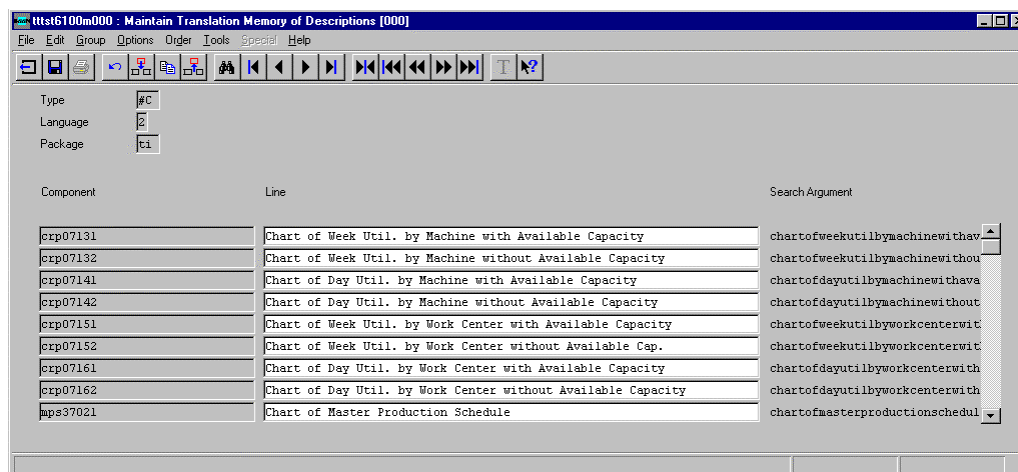
The Type codes in this session are identical to the component codes in the export files, with a few exceptions:

TM	Type	Export
#C	Chart descriptions	CO
CH	Chart options	Co
FO	Long and short form descriptions	FO/Fo
TI	Table indices	ID
QS	Questions	QT
RP	Report descriptions	RD

As you can see in the Search Argument field, all spaces, diacritics and punctuation are stripped from each string. The stripped strings in the source and the target language are compared in the Translate Descriptions by TM (ttadv6200m000) session, and if a match is found, a translation is generated.

10.2 Migrating a TM to a new version

The translation memory is stored in the following Tools tables: ttcon900, ttcon901 and ttcon902. These tables have to be copied to the new BAAN environment and then registered there. Existing translation memory data will be overwritten.

Steps :

- Read the tape (if applicable); it contains approx 400 MB of data.

- Copy the tables into the Tools data dictionary directory of the Baan environment. You can find the location of these tables by typing :

    `cat $BSE/lib/isamdef6.1 | grep ttcon`

- Register the tables by typing for each table:

    `tbase6.1 R r <full pathname>/<table name without extension>`

    Example: `tbase6.1 R r /usr2/BAAN40/000/tttcon/tcon902000`

The translation memory is now ready for use in the new environment.

10.3 Comparing package VRCs

If you want to find out about the net differences between package VRCs, go into Translation Utilities and start the Compare Package VRC's (ttadv6450m000) session. For software, fill out and check the fields as specified below. For help texts, check Help Text Codes and leave all other element fields unchecked.

The result is a listing of all elements that are either new or changed under the highest VRC specified.

Example:

New in B40 c sys2:

New	| mcs1543s000	|	| B40 c sys2 | Display States
New	| mcs1546m000	|	| B40 c sys2 | Display Exceptions for Taxes - Purchase
New	| mcs1546s000	|	| B40 c sys2 | Display Exceptions for Taxes – Purchase

Changed from B40 b to B40 c sys2:

Change	| qms0123s000	| B40 b	| B40 c sys2 | Maintain Algorithm Variables
Change	| qms0149s000	| B40 b	| B40 c sys2 | Maintain Order-Specific Inspections
Change	| qms1205s000	| B40 b	| B40 c sys2 | Check and Close Standard Inspections

11. Appendix A: UNIX and vi

11.1 Commands

This appendix addresses a number of UNIX commands. Each UNIX command is followed by a space and allows various options. Options are introduced with a hyphen (−) and can be combined into a string. Because this is not a UNIX course, please refer to a UNIX manual for more detail, especially regarding the options. Use all UNIX commands with caution, because there are no undo options.

When you log on to a UNIX machine, you will end up in a so-called home directory. To determine the name of that directory, or where you are at all in a UNIX directory structure, type:

pwd

then press ENTER. This command (print working directory) will yield the absolute path.

To change directory, type:

cd <directory>

Press ENTER. You can enter a new directory along the absolute path or the relative path. The absolute path is specified from the root directory (/); the relative path is specified from the current directory.

Absolute:**cd /usr5/baan/bse**

Relative: **cd ../../miscellaneous**

Type **cd** without a directory name and press ENTER to return to the home directory from anywhere in the directory tree.

To find out about the contents of a directory, type:

ls −l (on some machines, **ll** is an alternative spelling)

A typical file listing looks like this:

```
drwxrwxrwx 2 charles      bsp    320    Feb 5 16:34   accounts

drwxrwxrwx 2 charles      bsp    400    Feb 5 16:34   programs
```

Explanation:

- drwxrwxrwx
  The letter d designates a directory; file names start with a hyphen. The three rwx groups indicate file permissions. See section 1.2, Permissions.

- 2
  Link count. The number of names referring to the same data.

- charles
  Owner of the accounts directory.

- bsp
  Group for which the group permissions are valid.

- 320
  Directory size in bytes.

- Feb 5 16:34
  Date and time of creation or last update of the directory.

- accounts
  Directory name.

To create a directory, type:

   mkdir <directory>

Press ENTER. This command (make directory) will only create a directory if the name does not yet exist, and if the user has permission to make directories. This depends on the permissions of the directory where the command is issued.

To remove a directory, type

   rmdir <directory>

Press ENTER. This command (remove directory) will only remove a directory if it is empty.

To delete a file, type:

   rm <file>

Press ENTER.

To copy a file, type:

   cp <file> <destination>

Press ENTER. The destination can be another file, or a directory. If it is a directory, the user must have the appropriate permissions.

To rename a file, type:

   mv <file> <destination>

Press ENTER. The destination can be another file, or a directory. If it is a directory, the user must have the appropriate permissions.

To count lines, words and characters in a file, type

   wc [–lwc] <file>

Press ENTER. The –l option is for lines, –w, for words, and –c, for characters. **wc** without options shows all three.

This is how to do word counts for new versions and releases. For example:

1 for i in s*sch0
2 do
3 echo Counting $i…
4 echo $i `grep -v ^# $i | wc –l` > $i.cnt
5 done

Explanation:

Line 1: Assuming that there is a number of files in the current directory whose names
   start with s and end in sch0,
Line 2: execute the following actions.
Line 3: show which file is being processed.
Line 4: take all lines from that file that do **not** start with the number sign (#), count
   them and send the output to a file whose name is identical to the file being
   processed, with the extension .cnt.
Line 5: when there are no more files meeting the initial criterion, stop running.

To find character strings or words in a file, type:

   grep <string> <file>

Press ENTER.

grep stands for "Globally look for a Regular Expression and Print." The default output
is the screen; **grep <string> <file> > <output file>** sends the output to the specified
file.

Some useful options of the **grep** command:

–v (also known as 'negative grep', see the above example): show all lines that do not
   match the specified string.
–i (ignore case): no distinction is made between uppercase and lowercase.
–n (number): shows the line number in the file where the specified string occurs.

11.2   Permissions

UNIX allows the user to set access permissions for files. Directories are considered to
be files as well. Permissions are assigned on three levels: user, group, and others.
Consider the following result from the **ls –l** command:

```
-rwxr-xr-- 1   bill bsp   Feb 12 18:22    unix2pc
```

Bill is the owner of the file called unix2pc. Usually, he is also a member of the bsp
group. Other members from this group can read this file, members from other groups
cannot. The division is as follows:

```
rwxr-xr--
rwx  permissions for the owner
r-x  permissions for the group
r--  permissions for the others
```

The letters have the following meaning:

r readable
w writable
x executable

Bill, the owner of this file, can read, write, and execute it. His fellow members of the bsp group can only read and execute, but not write (that is, change) the file. All others can only read the file.

The owner of the file or the system manager (the UNIX super user, also known as root) can change the permissions by using the **chmod** command. There are two ways to issue this command: with binary digits and with letters.

11.2.1   Binary digits

For each group of three characters, eight settings are available.

1 --- no permission
2 --x execute
3 -w- write
4 -wx write and execute
5 r-- read
6 r-x read and execute
7 rw- read and write
8 rwx read, write, and execute

A 1 is used if an r, a w, or an x is set, a 0 is used if it is not. So --- translates as 000 and r-x, as 101. The entire permission setting for unix2pc translates into:

111   101   100
user   group   others

These are binary values. Their decimal counterparts are 7, 5, and 4. To set the permissions for unix2pc to rwxr-xr--, enter the following command:

> **chmod 754 unix2pc**

11.2.2   Letters

The user is defined as **u**, the group, as **g**, and the others, as **o**. Together they are defined as **a** (for all). To switch permissions (r, w, and x) on or off, + and – are used. To assign read and write permissions to user and group, and read permission to others, enter the following command:

> **chmod ug+rw,o+r unix2pc**

(do not use spaces in the permission argument).

Permissions apply to both files and directories, but they have a slightly different effect. For example, to copy to a directory, it must have write permission, and to be able to change to a directory, it must be executable. Consider the following table:

	file	directory
r	cat	ls
w	vi	cp, mv, rm
x	execute	cd, PATH

For the commands in this table that are not addressed in this appendix, please refer to your UNIX manual.

11.3 Using vi

vi (pronounced as two letters: *vee-ai*, and always written in lowercase) is a powerful UNIX text editor. It works with a command mode and an input mode. You can alternate between the two modes by pressing the ESC key. Start vi as follows:

vi <file name>

Press ENTER. If the file does not exist, it is created on the spot. Press ESC, **a** and start typing. If you want another command, press ESC again. Try and use vi and you will soon be familiar with the various options available.

11.3.1 Input commands

ESC	
a	add text at the cursor position
A	add text at the end of the line
i	insert text at the cursor position
I	insert text at the start of the line
o	add a blank line after the current line
O	add a blank line before the current line
r	overwrite the current character
R	overwrite text
C	change text to the end of the line
cw	change a word

11.3.2 Copy commands

ESC	
yy	copy the current line to a buffer
p	insert the buffer after the current line
P	insert the buffer before the current line

11.3.3   Delete commands

ESC	
dd	delete the current line
D	delete from the cursor position to the end of the line
dw	delete a word from the cursor position up to the first character of the next word
x	delete the character at the cursor position
X	delete the character before the cursor position

11.3.4   Find and replace commands

ESC	
/	find text
n	find next occurrence (forward search)
?	find previous occurrence (backward search)

:1,$s/<old string>/<new string>/g

from line 1 to the end of the file, substitute all occurrences of the old string with the new string. If you do not specify the **g** at the end, only the first occurrence on the line will be replaced. You can add a **c** at the end if you want to confirm each replace action.

11.3.5   Cursor movement

For up, down, left, and right, the arrow keys are available.

ESC	
w	go to the start of the next word
W	go to the start of the word after the next space
b	go to the start of the current word
B	go to the start of the preceding word
e	go to the end of the current word
E	go to the end of the next word
:1	go to line 1
:#	go to line #
:$	go to the last line
^	go to the start of the current line
$	go to the end of the current line

Many commands are repeatable. Try, for example, ESC, **15dd** to delete 15 lines, or ESC, **4cw** to change four words, or ESC, **5yy** to write 5 lines to a buffer, so you can copy them. Press ESC, followed by **u** to undo the last action. Refer to your vi manual for more details.